Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

Multitask matrix completion for learning protein interactions across diseases

Meghana Kshirsagar[1], Keerthiram Murugesan[2], Jaime G. Carbonell[2] and Judith Klein-Seetharaman[3]

[1]Memorial Sloan Kettering Cancer Center, 1275 York Ave., NY 10021
[2]Language Technologies Institute, Carnegie Mellon Univ., 5000 Forbes Ave., Pittsburgh PA 15213, USA
[3]Metabolic & Vascular Health, Warwick Medical School, Univ. of Warwick, Coventry, UK

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Disease causing pathogens such as viruses, introduce their proteins into the host cells where they interact with the host's proteins enabling the virus to replicate inside the host. These interactions between pathogen and host proteins are key to understanding infectious diseases. Often multiple diseases involve phylogenetically related or biologically similar pathogens. Here we present a multitask learning method to jointly model interactions between human proteins and three different, but related viruses: *Hepatitis C*, *Ebola* virus and *Influenza A*.

Results: Our multitask matrix completion based model uses a shared low-rank structure in addition to a task-specific sparse structure to incorporate the various interactions. We obtain between 7 to 39 percentage points improvement in predictive performance over prior state-of-the-art models. We show how our model's parameters can be interpreted to reveal both general and specific interaction-relevant characteristics of the viruses.

Availability: Code and data is available at: `http://www.cs.cmu.edu/~mkshirsa/bsl_mtl.tgz`

Contact: mkshirsa@cs.cmu.edu

1 Introduction

Infectious diseases such as H1N1 influenza, the recent Ebola outbreak and bacterial infections, such as the recurrent *Salmonella* and *E. coli* outbreaks are a major health concern worldwide, causing millions of illnesses and many deaths each year. Key to the infection process are host-pathogen interactions at the molecular level, where pathogen proteins physically bind to human proteins to manipulate important biological processes in the host cell, to evade the host's immune response and to multiply within the host. Very little is known about these protein-protein interactions (PPIs) between pathogen and host proteins for any individual disease. However, such PPI data is widely available across several diseases, and the central question in this paper is: *Can we model host-pathogen PPIs better by leveraging data across multiple diseases?* This is of particular interest for lesser known or recently evolved diseases where the data is particularly scarce. Furthermore, it allows us to learn models that generalize better across diseases by modeling global phenomena related to infection.

An elegant way to formulate the interaction prediction problem is via a graph completion based framework, where we have several bipartite graphs over multiple hosts and pathogens as illustrated in Figure 1. Nodes in the graphs represent host proteins (circles) and pathogen proteins (triangles), with edges between them representing interactions (host protein $\underline{interacts}$ pathogen protein). Given some observed edges (interactions obtained from laboratory based experiments), we wish to predict the other edges in the graphs. Such bipartite graphs arise in a plethora of problems including: recommendation systems (user $\underline{prefers}$ movie), citation networks (author $\underline{cites}$ paper), disease-gene networks (gene $\underline{influences}$ disease) etc. In our problem, each bipartite graph $\mathcal{G}$ can be represented using a matrix M, where the rows correspond to pathogen proteins and columns correspond to host proteins. The matrix entry M_{ij} encodes the edge between pathogen protein i and host protein j from the graph, with $M_{ij} = 1$

for the observed interactions. Thus, the graph completion problem can be mathematically modeled as a matrix completion problem (Candes and Recht, 2008).

Most of the prior work on host-pathogen PPI prediction has modeled each bipartite graph separately, and hence cannot exploit the similarities in the edges across the various graphs. Here we present a *multitask* matrix completion method that *jointly models* several bipartite graphs by sharing information across them. From the multitask perspective, a *task* is the graph between one host and one pathogen (can also be seen as interactions relevant to one disease). We focus on the setting where we have a single host species (human) and several related viruses, where we hope to gain from the fact that similar viruses will have similar strategies to infect and hijack biological processes in the human body. Such opportunities for sharing arise in other applications as well: for instance, predicting user preferences in movies may inform preferences in selection of books, or vice-versa, as movies and books are semantically related. Multitask learning based models that incorporate and exploit these correlations should benefit from the additional information.

Our multitask matrix completion based model is motivated by the following biological intuition governing protein interactions across diseases.

1. An interaction depends on the structural properties of the proteins, which are conserved across similar viruses as they have evolved from common ancestors. Our model thus needs a component to capture these latent similarities, which is *shared* across tasks.
2. In addition to the shared properties discussed above, each pathogen has also evolved specialized mechanisms to target host proteins. These are unique to the pathogen and can be expressed using a *task-specific* parameter in the model.

This leads us to the following model that incorporates the above ideas. The interactions matrix M_t of task t can be written as: $M_t = \mu_t *$ (shared component)$+(1-\mu_t)*$ (specific component), with hyperparameter μ_t allowing each task to customize its amount of shared and specific components.

To incorporate the above ideas, we assume that the interactions matrix M is generated from two components. The first component has low-rank latent factors over the human and virus proteins, with these latent factors jointly learned over all tasks. The second component involves a task specific parameter, on which we additionally impose a sparsity constraint as we do not want this parameter to overfit the data. Section 3 discusses our model in detail. We trade-off the relative importance of the two components using task-specific hyperparameters. Our model can thus learn what is conserved and what is different across pathogens, rather than having to specify it manually.

The key challenges in inducing such a model are: (1) In addition to the interactions from each graph, it should exploit information available in the form of features. (2) Exploiting features is particularly crucial since the graph $\mathcal{G}$ is often extremely sparse, i.e there are a large number of nodes and very few edges are observed. There will be proteins (i.e nodes) that are not involved in any known interactions – called the *cold start problem* in the recommendation systems community. The model should be able to predict the existence of links (or their absence) between such prior 'unseen' node pairs. This is of particular significance in graphs that capture biological phenomena. For instance, the host-pathogen PPI network of human-Ebola virus (column-3, Table 1) has $\approx$ 90 observed edges (equivalent to 0.06% of the possible edges) which involve only 2 distinct virus proteins. (3) A side-effect of having scarce data is the availability of a large number of unlabeled examples, i.e pairs of nodes with no edge between them. These unlabeled examples can contain information about the graph as a whole, and a good model should be able to use them.

The main contributions of this work are:

- We extend a prior matrix completion model (Abernethy *et al.*, 2009) to the multitask setting. This extension is new.
- Unlike most prior approaches, our model exploits node-based features which allows us to deal with the 'cold start' problem (generating predictions on unseen nodes).
- We apply the model to an important, real-world problem – prediction of interactions in disease-relevant host-pathogen protein networks, for multiple related diseases. We demonstrate the superior performance of our model over prior state-of-the-art multitask models.
- We use unlabeled data to initialize the parameters of our model, which serves as a prior. This gives us a modest boost in prediction performance.

1.1 Background: Host-pathogen protein interactions

The experimental discovery of host-pathogen protein interactions involves biochemical and biophysical methods such as co-immunoprecipitation (co-IP), yeast two-hybrid (Y2H) assays, co-crystalization. The host-pathogen protein interactions from several small-scale and high throughput experiments are aggregated by databases such as virus-MINTChatraryamontri *et al.* (2009), HPIDB Kumar and Nanduri (2010), PHISTO Tekir *et al.* (2012) etc by literature curation. These databases are valuable sources of information to bioinformaticians for developing models.

Prediction of host-pathogen PPIs: The most reliable experimental methods for studying protein-protein interactions (PPI) are often very time-consuming and expensive, making it hard to investigate the prohibitively large set of possible host-pathogen interactions – for example, the bacterium *Bacillus anthracis* which causes anthrax has about 2321 proteins which when coupled with the 100,000 or so human proteins gives $\approx$232 million protein pairs to test, experimentally. Computational techniques complement laboratory-based methods by predicting highly probable PPIs. These techniques use the known interactions data from previous experiments and predict the most plausible new interactions. In particular, supervised machine learning based methods use the few known interactions as training data and formulate the interaction prediction problem in a classification setting, with target classes: "interacting" or "non-interacting". Features are derived using various attributes of the two proteins such as: protein sequences from UniProt UniProt Consortium (2011), protein structure from PDB, gene ontology from GO database Ashburner *et al.* (2000).

1.2 Prior work

Most of the prior work in PPI prediction has focussed on building models separately for individual organisms (Chen and Liu, 2005; Wu *et al.*, 2006; Singh *et al.*, 2006; Qi *et al.*, 2006) or on building a model specific to a disease in the case of host-pathogen PPI prediction (Tastan *et al.*, 2009; Qi *et al.*, 2009; Dyer *et al.*, 2007; Kshirsagar *et al.*, 2012). There has been little work on combining PPI datasets with the goal of improving prediction performance

for multiple organisms. Qi *et al.* (2010) proposed a semi-supervised multi-task framework to predict PPIs from partially labeled reference sets. Kshirsagar *et al.* (2013) develop a task regularization based framework called MTPL that incorporates the similarity in biological pathways targeted by various diseases to couple multiple tasks together. Matrix factorization based protein-protein interaction (PPI) prediction has seen very little work, mainly due to the extremely sparse nature of these datasets which makes it very difficult to get reliable predictors. Xu *et al.* (2010) use a CMF-based approach in a multi-task learning setting for within species PPI prediction. The methods used in all prior work on PPI prediction do not explicitly model the features of the proteins and cannot be applied on proteins which have no known interactions available. Our work addresses both these issues.

A majority of the prior work in the relevant areas of collaborative filtering and link prediction includes single relation models that use neighbourhood based prediction (Sarwar *et al.*, 2001), matrix factorization based approaches (Koren *et al.*, 2009; Menon and Elkan, 2011) and bayesian approaches using graphical models (Jin *et al.*, 2002; Phung *et al.*, 2009). There have also been multitask approaches on link prediction (Zhang *et al.*, 2012; Cao *et al.*, 2010; Li *et al.*, 2009; Singh and Gordon, 2008) however many of those models do not incorporate features of the nodes in the graph. Menon and Elkan (2011) combine linear and bilinear features, latent parameters on the nodes and several other parameters into a function that minimizes a ranking loss for link prediction in a single graph. Abernethy *et al.* (2009) cast the problem of matrix completion in terms of the abstract problem of learning linear operators. Their framework allows the incorporation of features and kernels. We extend their bilinear model for the multitask setting. There has been a lot of work on other low-rank models for multitask learning (Ando and Zhang, 2005; Ji and Ye, 2009; Chen *et al.*, 2012, 2013). Another related line of work is co-embedding, where the goal is to embed different types of objects in the same shared lower-dimensional space (Mirzazadeh *et al.*, 2015).

2 Bilinear low-rank matrix decomposition

In this section, we present the matrix decomposition model that we extend for the multitask scenario. In the context of our problem, at a high level, this model states that – protein interactions can be expressed as dot products of features in a lower dimensional subspace.

Let $\mathcal{G}_t$ be a bipartite graph connecting nodes of type υ with nodes of type ς. Let there be m_t nodes of type υ and n_t nodes of type ς. We denote by $M \in \mathbb{R}^{m_t \times n_t}$, the matrix representing the edges in $\mathcal{G}_t$. Let the set of observed edges be Ω. Let $\mathcal{X}$ and $\mathcal{Y}$ be the feature spaces for the node types υ and ς respectively. For the sake of notational convenience we assume that the two feature spaces have the same dimension d_t [1]. Let $\mathbf{x}_i \in \mathcal{X}$ denote the feature vector for a node i of type υ and $\mathbf{y}_j \in \mathcal{Y}$ be the feature vector for node j of type ς. The goal of the general matrix completion problem is to learn a function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that also explains the observed entries in the matrix M. We assume that the function f is bilinear on $\mathcal{X} \times \mathcal{Y}$. This bilinear form was first introduced by Abernethy *et al.* (2009) and takes the following form:

$$f(\mathbf{x}_i, \mathbf{y}_j) = \mathbf{x}_i^\intercal H \mathbf{y}_j = \mathbf{x}_i^\intercal U V^\intercal \mathbf{y}_j \tag{1}$$

The factor $H \in \mathbb{R}^{d_t \times d_t}$ maps the two feature spaces $\mathcal{X}$ and $\mathcal{Y}$. This model assumes that H has a low-rank factorization given by $H = UV^\intercal$, where $U \in \mathbb{R}^{d_t \times k}$ and $V \in \mathbb{R}^{d_t \times k}$. The factors U and V project the two feature spaces to a common lower-dimensional subspace of dimension k. While the dimensionality of the feature spaces $\mathcal{X}$ and $\mathcal{Y}$ may be very large, the latent lower dimensional subspace is sufficient to capture all the information pertinent to interactions. To predict whether two new nodes (i.e nodes with no observed edges) with features $\mathbf{p}_i$ and $\mathbf{q}_j$ interact, we simply need to compute the product: $\mathbf{p}_i U V^\intercal \mathbf{q}_j$. This enables the model to avoid the cold start problem that many prior models suffer from. The objective function to learn the parameters of this model has two main terms: (1) a data-fitting term, which imposes a penalty for deviating from the observed entries in Ω and (2) a low-rank enforcing term on the matrix H.

The first term can be any loss function such as squared error, logistic-loss, hinge loss. We tried both squared error and logistic-loss and found the performance to be similar. The squared error function has the advantage of being amenable to adaptive step-size based optimization which results in a much faster convergence. The low-rank constraint on H (mentioned in (2) above) is NP-hard to solve and it is standard practice to replace it with either the trace-norm or the nuclear norm. Minimizing the trace norm (i.e. sum of singular values) of $H = UV^\intercal$, is equivalent to minimizing $\|U\|_F^2 + \|V\|_F^2$. This choice makes the overall function easier to optimize:

$$\begin{aligned} \mathcal{L}(U, V) \;&=\; \sum_{(i,j)\in\Omega} c_{ij}\; \ell(M_{ij}, \mathbf{x}_i^\intercal U V^\intercal \mathbf{y}_j) + \lambda(\|U\|_F^2 + \|V\|_F^2) \\ &\text{where } \ell(a, b) = (a - b)^2 \end{aligned} \tag{2}$$

The constant c_{ij} is the weight/cost associated with the edge (i, j) which allows us to penalize the error on individual instances independently. The parameter λ controls the trade-off between the loss term and the regularizer.

3 The bilinear sparse low-rank multitask model (BSL-MTL)

In the previous section, we described the bilinear low-rank model for matrix completion. Note that in order to capture linear functions over the features, we introduce a constant feature for every protein (i.e $[\mathbf{x}_i 1]$). We now discuss the multitask extensions that we propose. Let $\{\mathcal{G}_t\}$ where $t = 1 \ldots T$ be the set of T bipartite graphs and the corresponding matrices be $\{M_t\}$. Each matrix M_t has rows corresponding to node type υ_t and columns corresponding to the node type ς_t. The feature vectors for individual nodes of the two types be represented by $\mathbf{x}_{ti}$ and $\mathbf{y}_{tj}$ respectively. Let Ω_t be the set of observed links (and non-links) in the graph $\mathcal{G}_t$. Our goal is to learn individual link prediction functions f_t for each graph. In order to exploit the relatedness of the T bipartite graphs, we make some assumptions on how they share information. We assume that each matrix M_t has a low-rank decomposition that is shared across all graphs and a sparse component that is specific to the task t. That is,

$$f_t(\mathbf{x}_{ti}, \mathbf{y}_{tj}) = \mathbf{x}_{ti}^\intercal H_t \mathbf{y}_{tj}, \text{ where } H_t = \mu_t U V^\intercal + (1 - \mu_t) S_t \tag{3}$$

[1] the dimensions being different does not influence the method or the optimization in any way

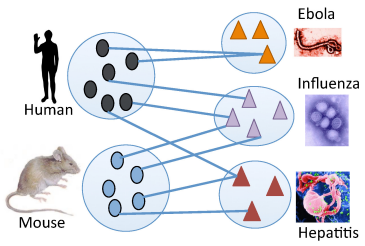

Fig. 1. Multiple bipartite graphs: on the left are host proteins and on the right are proteins from different virus species. Edges represent protein interactions.

Task →	*Influenza A* (28 strains)	*Hepatitis C* (5 strains, many isolates)	*Ebola* (3 strains)
number of HP PPIs (positives)	848	981	90
number of non-homologous PPIs	274	270	75
# of unique virus proteins in dataset	54	151	2
# of unique human proteins in dataset	362	385	88
total # of virus proteins in UniProtKB across strains †	542	163	38
number of negatives	84800	98100	9000
density of observed graph‡ (as %)	.15	.60	.20

HP PPI: host-pathogen protein protein interactions

‡ Density = (no. of positives + no. of negatives)/ total number of possible edges. The denominator was computed considering all proteins from all strains of the virus and ≈ 100,000 human proteins.

† this is the number of 'reviewed' proteins on UniProtKB from all strains in our data

Table 1. Tasks and their sizes (a column corresponds to one bipartite graph between human proteins and the virus indicated in the column header). More importantly, each task represents several strains of one virus species. We pool together PPI from all strains as the number of interactions per strain is very small and not representative. PPI data from different strains often involves homologous proteins (virus proteins or/and human proteins) as these were curated from different publications/studies involving the virus. The second row gives a conservative estimate of the number of non-homologous PPI. These were obtained by computing BLAST sequence similarity (see Section 4.1 for details). All pathogens are single stranded RNA viruses. The last row shows that each of our graphs is extremely sparse.

As before, the shared factors U and V are both $\mathbb{R}^{d_t \times k}$ (where the common dimensionality d_t of the two node types is assumed for convenience). The matrix $S_t \in \mathbb{R}^{d_t \times d_t}$ is a sparse matrix. The objective function for the multitask model is given by:

$$\mathcal{L}(U, V, \{S_t\}) = \frac{1}{N} \sum_{t=1}^{T} \sum_{(i,j)\in\Omega_t} c_{ij}^t \, \ell\big(M_{t_{ij}}, \mathbf{x}_{ti}^\intercal H_t \mathbf{y}_{tj}\big) + \lambda(\|U\|_F^2 + \|V\|_F^2) + \sum_{t=1}^{T} \sigma_t \|S_t\|_1 \quad (4)$$

Here $N = \sum_t |\Omega_t|$, is the total number of training examples (links and non-links included) from all tasks. To enforce the sparsity of S_t we apply an ℓ_1 norm. In our experiments, we tried both ℓ_1 and ℓ_2 norms and found that the ℓ_1 norm works better.

Optimization: The function $\mathcal{L}(U, V, \{S_t\})$ is non-convex. However, it is convex in every one of the parameters (i.e when the other parameters are fixed) and a block coordinate descent method called alternating least squares (ALS) is commonly used to optimize such functions. To speed up convergence we use an adaptive step size. The detailed optimization procedure is shown in the supplementary.

Convergence: The ALS algorithm is guaranteed to converge only to a local minimum. There is work showing convergence guarantees to global optima for related simpler problems, however the assumptions on the matrix and the parameter structure are not very practical and it is difficult to verify whether they hold for our setting.

Initialization of U and V: We tried random initialization (where we randomly set the values to lie in the range [0 1]), and also the following strategies that initialize: $U^0 \leftarrow$ top-k left singular vectors, and $V^0 \leftarrow$ top-k right singular vectors from the SVD of $\sum_{(i,j)\in\Gamma} m_{ij} \mathbf{x}_i \mathbf{y}_j^\intercal$. We set Γ to (a) training examples from all tasks, or (b) a random sample of 10000 unlabeled data from all tasks. We found that using the unlabeled data for initialization gives us a better performance.

3.1 Handling the 'curse of missing negatives'

For the MC algorithm to work in practice the matrix entries M_{ij} should represent interaction scores (range [0 1]) or take binary values (1s for positives and 0s for negatives). Our experiments with PPI probabilities (obtained using the MINT-scoring algorithm) gave bad models. The binary matrix setting requires some observed 0s. However non-interactions are not available as they cannot be verified experimentally for various reasons. Hence we derived a set of 'probable negatives' using a heuristic often used in PPI prediction work (Qi *et al.*, 2006, 2009; Dyer *et al.*, 2011; Kshirsagar *et al.*, 2013). We pair up all virus proteins with all human proteins and sample a random set to be negatives. This heuristic works in practice as the interaction ratio (i.e number of positives in a large random set of protein pairs) is expected to be very low: ≈ 1/100 to 1/500. That is, the probability that our negatives contain true positives is negligible.

High class imbalance: We incorporate the prior on the interaction ratio by setting the size of our randomly sampled negatives set equal to 100 times the number of gold standard positives.

4 Dataset

We use three human-virus PPI datasets from the PHISTO (Tekir *et al.*, 2012) database (version from 2014), the characteristics of which are summarized in Table 1. The *Influenza A* task includes various strains of flu such as: H1N1 (Puerto Rico, Texas strains etc), H3N2 (England strain etc), H5N1, H7N3. The *Hepatitis* task has several subtypes (1a, 1b), isolates such as H77, HC-J6 etc and *Ebola* has the Zaire ebolavirus strain Mayinga and strain 1995. All three are single-strand RNA viruses, with *Hepatitis* being a positive-strand ssRNA whereas *Influenza* and *Ebola* are negative-strand viruses. The density of the known interactions is quite small when considering the entire proteome (i.e all known proteins) of the host and pathogen species (last row in Table 1).

4.1 Addressing the homologs in PHISTO

The interactions data reported in PHISTO database is aggregated from several host-pathogen interactions data sources such as

MINT, IntAct, DIP etc. We found that many of the interactions (within a task) involve homologous PPI - virus protein homologs from two strains interacting with either the same human protein or with homologous human proteins. Such PPI usually come from different publications/ studies, but they introduce a lot of redundancy in the dataset. Removing such PPI is challenging as there are several criteria to consider, the most important being: *what threshold of sequence similarity constitutes a homolog*? The very existence of sequence similarity between proteins is what makes it possible for biologists to predict unknown interactions and most computational methods rely on it for PPI prediction.

In row-2 of Table 1, we give a conservative estimate of the number of non-homologous PPI. For this, we identify homologous PPI and retain only one of them in the dataset. Homologs were obtained using BLAST sequence alignment (blastp) using an e-value cut-off threshold of $1e^{-5}$. This is a relaxed cut-off since there are cases where two functionally different proteins (either virus-virus or human-human) with low query coverage and low identity end up as 'homologs'. For example, human proteins Q8WV28 (gene BLNK, B-cell linker protein) and O00459 (gene PIK3R2, Phosphatidylinositol 3-kinase) are considered homologs by this cut-off. The number of non-homologous PPI should therefore be considered a lower limit.

In our experiments section, we thus evaluate our models in two settings - (1) on the original dataset (2) by creating partitions: the first without any homologous PPI and the other with only the homologous PPI.

4.2 Features

Since the sequence of a protein determines its structure and consequently its function, it may be possible to predict PPIs using the amino acid sequence of a protein pair. Shen *et al.* (2007) introduced the "conjoint triad model" for predicting PPIs using only amino acid sequences. They partitioned the twenty amino acids into seven classes based on their electrostatic and water affinities.[2] A protein's amino acid sequence is first transformed to a class-sequence (by replacing each amino acid by its class). For k=3, they count the number of times each distinct tri-mer (set of three consecutive amino acids) occurred in the sequence. Since there are 343 (7^3) possible tri-mers (with an alphabet of size 7), the feature vector containing the tri-mer frequency counts will have 343 elements. To account for protein size, they normalized the counts by linearly transforming them to lie between 0 and 1. Thus the value of each feature in the feature vector is the normalized count for each of the possible amino acid three-mers. We use di-, tri- and four-mers thus leading to a total of 2793 features ($7^2 + 7^3 + 7^4$). Such features have been successfully applied in prior work (Dyer *et al.*, 2007; Kshirsagar *et al.*, 2013).

5 Experimental setup

5.1 Comparison with other machine learning approaches

Our baselines include recent low-rank and sparse models, conventional multitask methods and prior work on HP PPI prediction. For a uniform comparison we used least squared loss in all the methods. The MALSAR Zhou *et al.* (2011) package was used to implement some of the models. For the baselines wherever appropriate, we concatenated the features of the two node types into a single feature vector. Let $W \in \mathbb{R}^{T \times d_t}$ be the matrix with the task-specific weight vectors $\mathbf{w}_t$.

Single task (STL): We used ridge regression with ℓ_2 regularization (which performed better than ℓ_1).

MMTL: The mean regularized multitask learning model from Evgeniou and Pontil (2004).

Sparse + low-rank (Chen *et al.*, 2012): W is assumed to have the decomposition: $W = P + Q$, where P is sparse and Q has a low-rank structure.

IMC (Jain and Dhillon, 2013; Natarajan and Dhillon, 2014): This is the link-prediction model from Section 2, where data from all tasks is combined without incorporating any task relationships. U and V are shared by all tasks. We use the same initialization for this method as we do for our model. A comparison to this model tells us how much we gain from the task-specific sparsity component S_t.

MTPL (Kshirsagar *et al.*, 2013): A biologically inspired regularizer is used to capture task similarity.

BSL-MTL: Bilinear sparse low-rank multitask learning, the method developed in this paper.

5.2 Comparison with a homolog baseline

In addition to the machine learning methods presented in Section 5.1, we also tried a simple homology based approach. As per this method, a virus protein v will interact with a human protein h if it interacts with any other human protein h' similar to h. The converse holds too. Such homology-motivated heuristic approaches have been traditionally popular in the inter-organism PPI prediction literature (Walhout and Vidal, 2001; Matthews *et al.*, 2001). Mika and Rost (2006) offers a commentary on the the success of such approaches in prediction PPI within an organism and the transferability across organisms. Modifications of such approaches have been published recently (Chen *et al.*, 2009; Lin *et al.*, 2013; Murakami and Mizuguchi, 2014).

The virus-virus and human-human protein similarity graphs used in this baseline were computed using protein sequence similarity. We ran BLAST sequence alignment (`blastp`) with an e-value cut-off of 0.1 and the negative logarithm of the e-value was used as a measure of similarity (we use a somewhat relaxed cut-off as we are comparing across organisms). We use the entire human-virus interactome from PHISTO as input to this method. Given a human protein h, we rank all virus proteins v based on the following simple scheme. Let $V(h)$ be a set of all virus proteins interacting with the human protein h and let $H(v)$ be all human proteins interacting with the virus protein v. We score each pair (h, v) from the bipartite graph as follows: $\max\{\max$ similarity between h and all human proteins in $H(v)$; $\max$ similarity between v and all virus proteins in $V(h)\}$. This baseline is shown in the first row as *Homolog* in Table 2.

5.3 Evaluation setup

We compare all the methods in two settings, where a small proportion of the available labeled data is randomly sampled and used to train a model which is then evaluated on the remaining data. For the first setting we randomly split the labeled data from each task into 10% training and 90% test, such that the class-skew of 1:100 is maintained in both splits (i.e stratified splits). The second setting uses a 30% training, 70% test split. In each setting we generate ten random splits and average the performance over the ten runs.

[2] For details of these classes, please refer to the supplementary or the original paper

We report the area under the precision recall curve (AUC-PR) along with the standard deviation. AUC-PR has been shown to give a more informative picture of an algorithm's performance than ROC curves in high class imbalance datasets (Davis and Goadrich, 2006) such as ours.

Parameter tuning: Please refer to the supplementary.

6 Results

The PPI data we obtain from PHISTO has homologous PPI within a task. Note that the presence of sequence similarity across virus proteins is a strong criterion for computational methods to work, however orthologs or paralogs of proteins within a task lead to redundancy in the data and some readers may find it harder to judge the contribution of the methods. We therefore present results on three sets:

1. the original PPIs from PHISTO. AUC-PR is shown in Table 2
2. **Non-homologous partition**: after removal of all homologous PPI from the test data. Here we ensure that the test data does not contain homologs of PPI observed in the training data (this ensures there are no redundant examples). Further, every set of homologous PPIs in the test data is replaced with a single PPI. A BLAST sequence similarity cut-off of $1e^{-5}$ was used to find homologs. The results are in Table 3.
3. **Homologous PPI partition** : only homologous PPI part of the test data. This is essentially the set of PPI removed in the set-(b) above. Table 4 shows these results.

Note that the partitions in (b) and (c) together comprise our test data. Also, the negatives in the test data are randomly split into the 'no-homolog' and 'homolog' partitions while maintaining the number of negatives in each partition to be 100 times the number of positives.

Overall performance For reference, the AUC-PR of a random classifier model is ≈ 0.01 due to the class imbalanced nature of the data. In general, we notice that multitask learning benefits all tasks. The first three columns show the results in the 10% setting. Our model (last row) has large gains for Influenza (1.4 times better than the next best) and modest improvements for the other tasks. The variance in the AUC is high for the Ebola task (column 1) owing to the small number of positives in the training splits. The most benefits for our model are seen in the 30% setting for all tasks. Notably, Homolog performs comparably to our model. In addition to the training data seen by all other methods, this baseline also uses additional data in the form of the virus-virus, human-human sequence similarities and a virus-human interactome from several other viruses (besides the three tasks). Results on the partitions shed further light on this.

Performance on the non-homologous PPI partition: Table 3 has these results. Overall, our method performs best on this partition over all tasks. Comparison with the AUC in Table 2 shows that AUC for the *Ebola* task is higher on all baselines except Homolog, indicating that a bigger fraction of the overall performance comes from the non-homologous partition (this is also marginally the case for *Influenza*). This effect is the most pronounced for our method (BSL-MTL) and demonstrates that we are able to predict such PPI by sharing information with other tasks. There are thus two components to the prediction performance - (1) the presence of sequence-similar proteins within a task (2) non-linear similarities between the interactions within a task and across tasks. We are able to capture the latter well via the shared subspace structure of our model.

10 % training, 90% test on Non-homologous PPI			
	Ebola	*Hep-C*	*Influenza*
Homolog	0.207 ±.07	0.113 ±.01	0.109 ± .01
STL	0.210 ±.14	0.474 ± .06	0.302 ±.03
Sparse+LR	0.216 ±.13	0.482 ±.05	0.312 ±.04
BSL-MTL	**0.295** ±.10	**0.544** ±.06	**0.505** ±.05

Table 3. AUC-PR on the non-homologous PPI within each task, in the 10% setting. To get an estimate of the relative size of the non-homologous partition for each task, see Table 1. Only the most representative baselines are retained for this experiment. Our model performs better than the others on the non-homologous part of the test data, as it captures non-linear similarities among the interactions within a task and across tasks very well.

Performance on the homologous PPI partition: Except the *Ebola* task, BSL-MTL has the highest AUC. The Homolog baseline best captures the homologous PPI in *Ebola* (with an AUC twice the next best). However, since there are far fewer homologous PPI in Ebola than in the other tasks (row-2 of Table 1), this contributes less to the overall performance of the Homolog baseline. *Hepatitis-C* has the highest fraction of homologous PPI and BSL-MTL does significantly better on these than the other baselines which also leads to a higher overall AUC. The same can be said of the *Influenza* task.

	10% training, 90% test			**30% training, 70% test**		
	Ebola	*Hep-C*	*Influenza*	*Ebola*	*Hep-C*	*Influenza*
Homolog	**0.230**±.06	0.178±.01	0.158±.01	**0.311**±.03	0.180±.01	0.198±.01
STL (Ridge Reg.)	0.189±.09	0.702±.08	0.286±.02	0.130±.03	0.802±.03	0.428±.03
MMTL (Evgeniou and Pontil, 2004)	0.113±.04	0.767±.03	0.321±.02	0.129±.02	0.802±.04	0.430±.03
Sparse+low-rank (Chen *et al.*, 2012)	0.144±.07	0.767±.02	0.318±.02	0.153±.02	**0.814**±.01	0.414±.03
MTPL (Kshirsagar *et al.*, 2013)	0.217±.08	0.695±.02	0.345±.02	0.260±.05	0.713±.01	**0.496**±.03
IMC (Natarajan and Dhillon, 2014)	0.087±.04	**0.779**±.02	**0.362**±.01	0.122±.02	0.801±.01	0.410±.03
BSL-MTL (this work)	**0.233**±.10	**0.807**±.02	**0.486**±.02	**0.361**±.03	**0.842**±.01	**0.560**±.02

Table 2. Area Under the Precision-Recall curve (AUC-PR) on the PHISTO dataset. The column header 'X% training' indicates the fraction of the labeled data used for training and tuning the model with the rest (100-X)% used as test data. We report the average AUC-PR over 10 random train-test splits (stratified splits that maintain the class-skew of 1:100). The standard deviation is also shown. The performance of the best baseline and the overall best method (BSL-MTL) is highlighted in bold. The Homolog baseline performs comparably to our BSL-MTL method on the Ebola task. This is explained by the virus-human PPI graph, from several other viruses, that the Homolog baseline has access to.

10 % training, test only Homologous PPI			
	Ebola	*Hep-C*	*Influenza*
Homolog	0.399 ±.07	0.161 ±.01	0.180 ± .01
STL	0.192 ±.15	0.504 ±.10	0.305 ±.03
Sparse+LR	0.184 ±.13	0.512 ±.10	0.315 ±.04
BSL-MTL	0.130 ±.10	0.815 ±.02	0.475 ±.02

Table 4. AUC-PR on the homologous PPI within each task, in the 10% setting. The Homolog baseline outperforms all on the Ebola task and BSL-MTL beats all other methods on the other tasks. The Homolog method has access to a wider virus-human PPI graph involving several other viruses and is able to infer homologous PPI for Ebola better (note that some of the homologous PPI are not redundant examples but involve proteins with very small regions of sequence similarity.

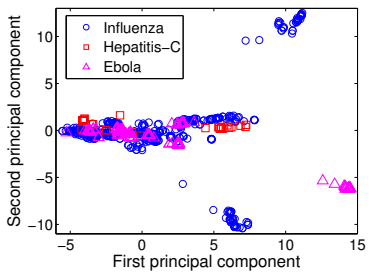

Fig. 2. Principal component analysis (PCA) of virus proteins in the original feature space. The first two principal components are shown. Shape and color of the points indicates which virus that protein comes from.

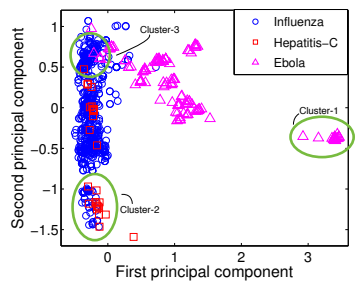

Fig. 3. Principal component analysis (PCA) of virus proteins in the projected subspace. The first two principal components are shown. Shape of the points indicates which virus that protein comes from. We have highlighted three clusters for which we show GO term enrichment analysis results in Table 6.1.1.

Gene Ontology terms for clusters from Figure 3

cluster 1 (ebola only)	mRNA (guanine-N7-)-methyltransferase activity; RNA-directed RNA polymerase activity; ATP binding
cluster 2 (hepatitis & influenza proteins)	host cell endoplasmic reticulum membrane; host cell lipid particle; induction by virus of host autophagy; host cell mitochondrial membrane; suppression by virus of host STAT1 activity
cluster 3 (all)	fusion of virus membrane with host plasma membrane

Table 5. GO terms (for process, function, component) enriched in the highlighted clusters from Figure 3. FuncAssociate GO enrichment tool was used and terms with p-value < 0.001 are shown. The first column also indicates the composition of each cluster (i.e which task's proteins appear in the cluster).

6.1 Biological significance of the model

The model parameters U, V and S are a source of rich information which can be used to further understand host-pathogen interactions. Note that our features are derived from the amino acid sequences of the proteins which provide opportunities to interpret the parameters.

6.1.1 Clustering proteins based on interaction propensities

We analyze the proteins by projecting them using the model parameters U and V into a lower dimensional subspace (i.e computing $XU^\intercal$ and $YV^\intercal$ to get projections of the virus and human proteins respectively). The principal component analysis (PCA) of this lower dimensional representation is compared with PCA in the original feature space (protein sequence features) in Figures 2 and 3. Firstly, the projected data has a much better separation than the original data. Secondly, Fig 3 tells us that Hepatitis-C and Influenza have many proteins with similar binding tendencies, and that these behave differently than most Ebola virus proteins. This observation is not obvious in the PCA of the original feature space (Fig 2), where proteins with similar sequences group together. We analyze the projected data further by looking at the clusters of proteins for enrichment of Gene Ontology (GO) annotations (proteins were first clustered in this lower dimensional space using k-means and setting the number of clusters to 6). Of particular interest are the highlighted three clusters which contain either proteins projected far from others (such as cluster-1) and proteins from different viruses projected close together (cluster-2 and cluster-3). For enrichment analysis, we use the FuncAssociate 3.0 (Berriz *et al.*, 2003) tool and GO annotations for the three viruses from UniProtKB.

6.1.2 Novel interactions with Ebola proteins

The top four Ebola-human PPI are all predictions for the Ebola envelope glycoprotein (GP) with four different human proteins (Note: GP is not in the gold standard PPIs). There is abundant evidence in the published literature (Nanbo *et al.*, 2010) for the critical role played by GP in virus docking and fusion with the host cell.

6.1.3 Sequence motifs from virus proteins

We analyze sequence motifs derived from the top k-mers that contribute to interactions. The significant entries of the model parameters U, V and $\{S_t\}$ were used to compute these motifs. The top positive-valued entries from the product UV^T indicate which pairs of features: $((f_v, f_h)$: virus protein feature, human protein feature) are important for interactions across all the virus-human PPI tasks. Analogously, the entries from S_t give us pairs of features important to a particular virus-human task 't'. We find that most of the top entries from UV^T correspond to linear virus features, whereas those from the various S_t involve bilinear features. These results are in the supplementary.

6.2 Results on HIV data

We incorporate an additional task: HIV (human immunodeficiency virus), which is quite dissimilar to our original setting involving three viruses (HIV is a retrovirus). This experiment tests how our multitask model behaves in the presence of a relatively outlier task. The interactions data has ≈4000 human-HIV PPI across several strains of which 1320 are non-homologous PPI. Please refer to the supplementary for the results.

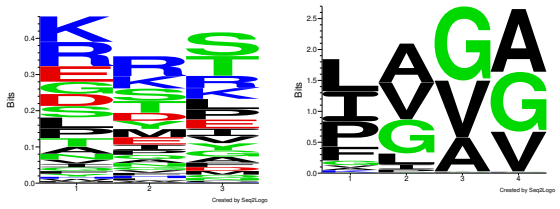

Fig. 4. (Left) Top tri-mer sequence motifs from virus proteins across all three viruses. These correspond to features important for interactions and is derived from our model parameter $H = UV^T$. (Right) Motif corresponding to enriched 4-mer patterns found by our model (parameter: S_{hepc}) for the Hepatitis-C virus task.

7 Conclusions and future extensions

This work developed and tested a new multitask learning method for host-pathogen PPI prediction based on low-rank matrix completion for sharing information across tasks. Our method, BSL-MTL exhibited large increases in prediction accuracy compared to a variety of baselines. The model parameters provide several avenues for further studying host-pathogen PPI that can lead to interesting observations and insights. Finally, the model we present is general enough to be applicable on other problems such as: gene-disease relevance prediction across organisms or disease conditions.

References

Abernethy, J., Bach, F., Evgeniou, T., and Vert, J. P. (2009). A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research (JMLR)*.

Ando, R. K. and Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, **6**, 1817–1853.

Ashburner, M. *et al.* (2000). Gene ontology: tool for the unification of biology. *Nat. Genet.*, **25**(1), 25–9. `http://www.geneontology.org/`.

Berriz, G., King, O., Bryant, B., *et al.* (2003). Characterizing gene sets with funcassociate. *Bioinf.*, **19**(18), 2502–04. `http://llama.mshri.on.ca/funcassociate/`.

Candes, E. and Recht, B. (2008). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*.

Cao, B., Liu, N. N., and Yang, Q. (2010). Transfer learning for collective link prediction in multiple heterogenous domains. *ICML*.

Chatraryamontri, A., Ceol, A., Peluso, D., *et al.* (2009). Virusmint: a viral protein interaction database. *Nucleic Acids Research*, **37**.

Chen, C.-C., Lin, C.-Y., Lo, Y.-S., and Yang, J.-M. (2009). Ppisearch: a web server for searching homologous protein–protein interactions across multiple species. *Nucleic acids research*, **37**(suppl 2), W369–W375.

Chen, J., Liu, J., and Ye, J. (2012). Learning incoherent sparse and low-rank patterns from multiple tasks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, **5**(4), 22.

Chen, J., Tang, L., Liu, J., and Ye, J. (2013). A convex formulation for learning a shared predictive structure from multiple tasks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **35**(5), 1025–1038.

Chen, X. and Liu, M. (2005). Prediction of protein-protein interactions using random decision forest framework. *Bioinformatics*, **21**(24), 4394–400.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. *Proceedings of the 23rd international conference on Machine learning*, pages 233–240.

Dyer, M. *et al.* (2011). Supervised learning and prediction of physical interactions between human and hiv proteins. *Infect., Genetics and Evol.*, **11**, 917–923.

Dyer, M., Murali, T., and Sobral, B. (2007). Computational prediction of host-pathogen protein-protein interactions. *Bioinf.*, **23**(13), i159–66.

Evgeniou, T. and Pontil, M. (2004). Regularized multi-task learning. *ACM SIGKDD*.

Hornbeck, P. V., Zhang, B., Murray, B., Kornhauser, J. M., Latham, V., and Skrzypek, E. (2015). Phosphositeplus, 2014: mutations, ptms and recalibrations. *Nucleic acids research*, **43**(D1), D512–D520.

Jain, P. and Dhillon, I. S. (2013). Provable inductive matrix completion.

Ji, S. and Ye, J. (2009). An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 457–464. ACM.

Jin, R., Si, L., and Zhai, C. (2002). Preference-based graphic models for collaborative filtering. In *UAI*, pages 329–336.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8), 30–37.

Kshirsagar, M., Carbonell, J. G., and Klein-Seetharaman, J. (2012). Techniques to cope with missing data in host-pathogen protein interaction prediction. *Bioinformatics*.

Kshirsagar, M., Carbonell, J. G., and Klein-Seetharaman, J. (2013). Multi-task learning for host-pathogen protein interactions. *Bioinformatics*.

Kumar, R. and Nanduri, B. (2010). Hpidb–a unified resource for host-pathogen interactions. *BMC Bioinf.*

Li, B., Yang, Q., and Xue, X. (2009). Transfer learning for collaborative filtering via a rating-matrix generative model. In *International Conference on Machine Learning*, pages 617–624. ACM.

Lin, C.-Y., Chen, Y.-C., Lo, Y.-S., and Yang, J.-M. (2013). Inferring homologous protein-protein interactions through pair position specific scoring matrix. *BMC bioinformatics*, **14**(2), 1.

Matthews, L. R., Vaglio, P., Reboul, J., Ge, H., Davis, B. P., Garrels, J., Vincent, S., and Vidal, M. (2001). Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or âĽœinterologsâĽž. *Genome research*, **11**(12), 2120–2126.

Menon, A. K. and Elkan, C. (2011). Link prediction via matrix factorization. *ECML*.

Mika, S. and Rost, B. (2006). Protein–protein interactions more conserved within species than across species. *PLoS Comput Biol*, **2**(7), e79.

Mirzazadeh, F., White, M., György, A., and Schuurmans, D. (2015). Scalable metric learning for co-embedding. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-KDD)*, pages 625–642. Springer.

Murakami, Y. and Mizuguchi, K. (2014). Homology-based prediction of interactions between proteins using averaged one-dependence estimators. *BMC bioinformatics*, **15**(1), 1.

Nanbo, A., Imai, M., Watanabe, S., *et al.* (2010). Ebolavirus is internalized into host cells via macropinocytosis in a viral glycoprotein-dependent manner. *PLoS pathogens*, **6**(9), e1001121.

Natarajan, N. and Dhillon, I. S. (2014). Inductive matrix completion for predicting geneâĽ“disease associations. *Bioinformatics*.

Phung, D. Q., Venkatesh, S., *et al.* (2009). Ordinal boltzmann machines for collaborative filtering. In *UAI*, pages 548–556. AUAI Press.

Qi, Y. *et al.* (2006). Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Proteins*, **63**(3), 490–500.

Qi, Y., Dhiman, H., Bar-Joseph, Z., *et al.* (2009). Systematic prediction of human membrane receptor interactions. *Proteomics*, **23**(9), 5243–55.

Qi, Y., Tastan, O., Carbonell, J. G., Klein-Seetharaman, J., and Weston, J. (2010). Semi-supervised multi-task learning for predicting interactions between hiv-1 and human proteins. *Bioinformatics*.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295. ACM.

Shen, J. *et al.* (2007). Predicting protein-protein interactions based only on sequences information. *PNAS*, **104**, 4337–4341.

Singh, A. P. and Gordon, G. J. (2008). Relational learning via collective matrix factorization. *KDD*.

Singh, R., Xu, J., and Berger, B. (2006). Struct2net: Integrating structure into protein-protein interaction prediction. *Pacific Symposium on Biocomputing*.

Tastan, O. *et al.* (2009). Prediction of interactions between hiv-1 and human proteins by information integration. *Pac. Symp. Biocomput.*, (14), 516–527.

Tekir, S. D., S., A., C., T., and U., K. O. (2012). Infection strategies of bacterial and viral pathogens through pathogen-host protein protein interactions. *Frontiers in Microbial Immunology*.

Thomsen, M. C. F. and Nielsen, M. (2012). Seq2logo: a method for construction and visualization of amino acid binding motifs and sequence profiles including sequence weighting, pseudo counts and two-sided representation of amino acid enrichment and depletion. *Nucleic acids research*, **40**(W1), W281–W287.

UniProt Consortium (2011). Ongoing and future developments at the universal protein resource. *Nucl. Acids Res.*, **39**, D214–D219.

Vita, R., Overton, J. A., Greenbaum, J. A., *et al.* (2015). The immune epitope database (iedb) 3.0. *Nucleic acids research*, **43**(D1), D405–D412.

Walhout, A. J. and Vidal, M. (2001). Protein interaction maps for model organisms. *Nature Reviews Molecular Cell Biology*, **2**(1), 55–63.

Wu, X., Zhu, L., Guo, J., *et al.* (2006). Prediction of yeast protein-protein interaction network: insights from the gene ontology and annotations. *Nucleic Acids Res.*, **34**(7), 2137–50.

Xu, Q., Xiang, E. W., and Yang, Q. (2010). Protein-protein interaction prediction via collective matrix factorization. *International Conference on Bioinformatics and Biomedicine*.

Zhang, Y., Cao, B., and Yeung, D.-Y. (2012). Multi-domain collaborative filtering. *arXiv preprint arXiv:1203.3535*.

Zhou, J., Chen, J., and Ye, J. (2011). Malsar: Multi-task learning via structural regularization. *Arizona State University*.